UASFunPolice

Release 0.0

Chris Gough

February 17, 2015

CHAPTER 1

Introduction

The purpose of this document is to improve safety of CanberraUAV.

It is a living document, hosted at https://github.com/monkeypants/UASFunPolice/ . Pull requests are welcome, as are comments/issues/suggestions through the github ticket system.

It's written "aspirationally" as though some software called UASFunPolice exists, however that's vapourware at this stage. In reality, the first iteration will be the simplest bodgy hack we can get away with (using existing systems), with a plan to improve iteratively from there...

Unmaned Aerial Systems (UAS) comprise some obvious hardware components like Unmanned Aerial Vehicles (UAVs) and physical components of the ground-segment, such as Ground Control Stations (GCSs). They also involve soft operational components, such as procedures, checklists and documention. Perhaps less obviously, they also involve soft management components, such as legal regulation, range safety rules, organisation policies, certifications, etc.

The most important management component of any aviation operation is it's Safety Management System (SMS). The purpose of these Safety Management Tools for Unmanned Aerial Systems (UASFunPolice) is to enable UAS operators to administer their own SMS to a standard that is comparable with manned aviation.

UASFunPolice is free (open source) software released under the Gnu Public Licence. The development team is engaged UAS Research and Development (R&D) for of Search and Rescue (SAR) applications. Despite our narrow focus, we hope that wider group of aviators might find these tools useful.

CHAPTER 2

Administration Requirements

We assume there will be administrative tasks that support operations and management. We are not really sure what they are yet...

2.1 Manage System Configuration

Basically, modifying the behavior of the software by changing settings. What software? What behavior? What settings? Who knows?

This is a supporting use-case package, it does not directly improve safety.

If we are changing the behavior of the system, we will require some sort of governance arangement (to analyse impacts of changes on users). This will also require record keeping that is auditable.

2.2 Manage Equipment

This is a placeholder, in anticipation that we will need to manage some information about the equipment we use. A supporting use-case package.

2.3 Manage People/Groups

The SMS is comprised of processes and systems to make the enterprise safer. Those systems are used by people. People participate in those processes (probably with explicit roles).

Manage people includes manage accounts / access control to the systems.

If we end up designing a multi-tennanted solution, we may well need a concept of “groups” as in, each person may belong to zero or more groups, and people can create/join/leave groups.

CHAPTER 3

Management Requirements

3.1 Promote Safety

Key points:

- Follow up on incidents, accidents and hazards by creating agenda items for group meetings (e.g. the Monday Mumble session).
- Publish, distribute and promote relevant artefacts.

CanberraUAV is a self-organising, non-profit group “staffed” by volenteers that work on open source research and development projects. We are a registerd community organisation with a bank account and a comittee (etc), but we do not have a command structure or hirarchy like a conventional organisation.

The cultural context of self-organised community groups is a significant factor in the way we can promote safety. There is very little scope for authoratitave mandates. The communication strategies most likely to succeed will involve the safety management system (and team) continuously demonstrating the merits of their activities.

3.2 Improve Safety

This is where we figure out what needs to be done and do it. It’s a continuous improvement, closed-loop feedback process.

It probably involves improving policies and procedures, initiating cultural change as well as other safety-improving actions.

One idea is to utilise a ticketing system (such as GitLab) for safety issues, which are referenced (or even closed) by changes to policy and procedure documentation in a version control system (such as git).

3.3 Assess Safety

Opreational data (including planned activity) is systematically reviewed, evaluated, discussed and analysed in a timely fashion. The results of this work is fed into safety improving and promoting activities.

CHAPTER 4

Operational Requirements

4.1 Plan Activity

All recorded activity occurs against some kind of plan. Some activities would be able to reuse generic plans. Other activities would need specific planning.

See www.ozrunways.com; preflight checking/planning for australian aviators. Check for NOTAMS, airspace, etc. Maybe even automated preflight checking?

4.1.1 Standard Operations

There is probably a standard set of generic procedures, and activities that comply with those procedures might not require aditional planning. They are performed at the operators discression and the activity is recorded after the fact. Changes to these procedures would be subject to safety assessment.

For example, CanberraUAV flys multiple missions most weeks at the CMAC airfield. These usually involve flight testing incremental changes to software, airframes or avionics. These flights comply with the range safety plan of the CMAC site, and the Standard Operating Procedures of the Model Aircraft Association of Australia. They are usually observed by a gallery of experienced aeromodellers and always flown by a suitably qualified safety pilot.

It would be difficult and disruptive to impose manditory planning steps to Standard Operations such as these.

4.1.2 Extraordinary Operations

Extraordinary activity is defined as anything outside the bounds of Standard Operations. These require advanced planning, which is subject to safety assessment prior to activity occuring. New standard operations would be subject to equivalent process as extraordinary activities.

The activity planning domain probably inclueds concepts like:

- range safety plan
- class of airspace
- class of activity

The scope of the risk assessment associated with activity planning is greater than safety; it also includes risks related to regulatory compliance, financial, reputation and others. We probably need to link up to a “risk management plan” framework of which safety management is a subset.

4.2 Record Activity

Activity is recorded in a number of contexts:

- workshop (construction/maintenance)

- flight planning
- packing and unpacking equipment from transport/storage
- pre-flight checking
- telemetry/telecommand/payload data
- post-flight checking
- communication logs
- incident/accident management

Some activities must be very simple (low effort) to record. For example, upload telemetry/telecommand log files along with the absolute minimum of information. To the maximum extend possible, this sort of information management should be handled automatically.

The complicated cases include all the incident, accident and hazard reporting features.

This will not be limited to flight operation activities. Inspection and maintenance of equipment. Maybe even our meeting minutes belong here. Much to elaborate on...

4.3 Submit a report to the Safety Team

Typically this would be done by a UAS operator, but the safety team would accept reports from anyone. In other words, it might be reported when flight logs were uploaded, or it might be reported independant of uploading flight logs.

4.3.1 Report a Hazard

A hazard is the potential for an incident or accident. The risk is percieved, the hazard report is simply an issue or problem for the safety team to evaluate, with some potential to improve safety.

The submitter may request that the Hazard report is treated in-confidence. In this situation, the safety team may disclose "lessons learned" and other topics related to the hazard, but keep the specifics of the hazard report private (for example, who reported it and exactly when).

4.3.2 Report an Incident

An incident is something that actually happened (at a time and place). Nobody was hurt, nothing was significantly damaged, but a percieved risk was validated by events.

The safety team will investigate every reported incident, using the same sort of analysis as used for more serious accidents. The investigation will generally be conducted internaly (within the Safety Team), without resorting to external parties. Where appropriate, incident investigations may be reviewed by a third party, for example by an aviation safety auditor.

Submitters may request that an incident report is kept confidential. In this case, the details of the incident report will be discussed among the safety team. It may also be shared with appropriate third parties, but it will not be released into the public domain. The safety team may disclose "lessons learned" and other non-specific details, but keep the specifics of the incident report private.

If a submitter nominates that they do not wish for the incident report to be kept private, the safety team may release it into the public domain at their discression,

4.3.3 Report an Accident

An accident is an incident with bad consequences. For example, personal injury significant damage to equipment.

The safety team will investigate all reported accidents. As appropriate, they will also forward the accident report to relevant parties and authorities. It may not be possible to agree to keep accident reports confidential, however if requested the safety team can assure maximum possible discretion (as oposed to discussing the accident openly).

4.4 Suppliment HIA reports with additional data

Hazards, Incidents and Accidents are reported using a standard form, becuase it prompts the submitter to provide certain details that are considered useful a-priori. Where available, it may be beneficial to include additional data to augment the information in the standard form. For example telemetry logs, video and still images, audio, diagrams, journalism references, additional witness statements, etc. Unlike the fields of the standard form, these are essentially unstructured data.

4.4.1 Directly attach media files to an HIA report

At the time a HIA report is submitted, the submitter may attach media files directly. For example, attach them to an email that they send to an HIA report submission inbox, or use upload features of the online HIA reporting tool. Where practical to do so, this would usually be the preferred method.

4.4.2 Link media to an HIA report

At the time a HIA report is submitted, the submitter may include hyperlink references to media hosted elseware, such as youtube videos or droneshare telemetry. Where this media is password protected, the sumitter would need to provide access credentials.

In some situations this might be the most practical way to provide supplimentary data, for example where a significantly large volume of data were involved, or where the origional source material is not available to the submitter. However, the downside of hyperlinks to remotely hosted data is that it may cease to be available at some point in the future, making future reviews or audits more difficult.

4.4.3 Provide supplimentary data after submission

Either at the request of the safety team or unprompted, a HIA report submitter may chose to augment a HIA report with supplimentary data after the report has been submitted. This may be linked media or directly attached files.

4.5 Anonymously Report Concerns

Anonymous reporting has a crucial role in aviation safety. The functional requirements are simple - anyone can report a hazard or incident (concern) anonymously, and these will be (at the very least) reviewed and considered by the safety management team.

If the incident involved loss of life or other very serious consequences, and the Australian Transport Safety Beuro (or equivalent authority in foreign jurisdictions) are required to investigate, then the safety team could be mandated by law to release all relevent information to the investigators. This probably means they need to be able to break anonaminity.

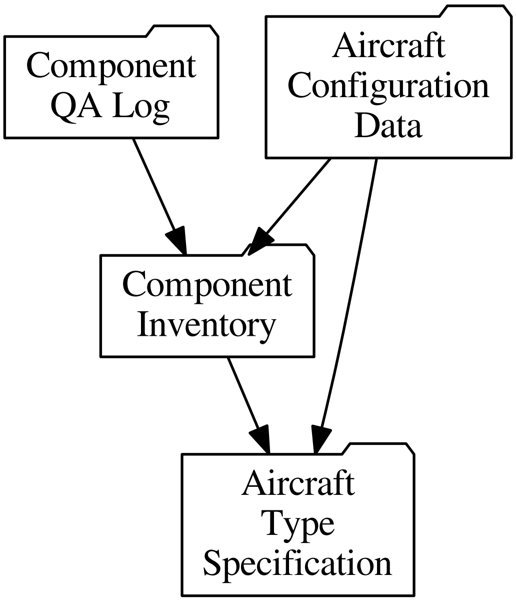
CHAPTER 5

Domain Model

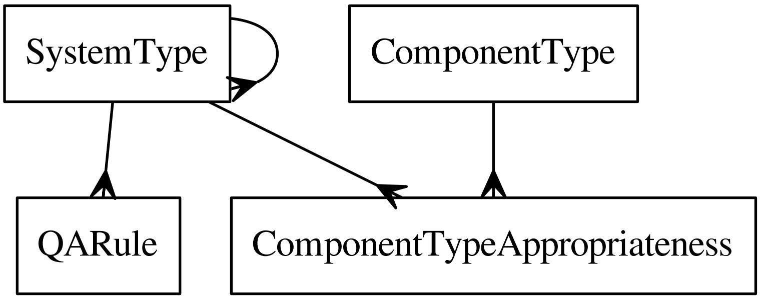
This blather was typed as part of a late night sprint, and requires a lot of rework before it's usefull. Best to skip it and focus on the requirements sections for the timebeing.

5.1 Vehicle Configuration

Overall picture (package diagram) of vehicle management database.



Aircraft Type Specification



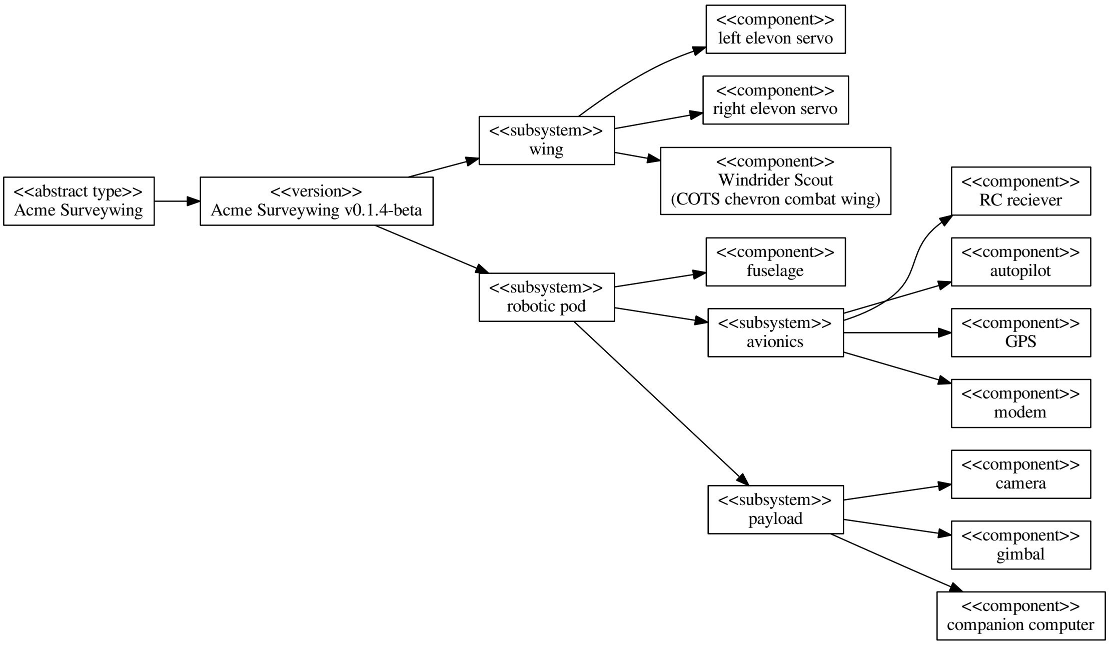
5.1.1 SystemType

The core of the Aircraft Type Specification is the **SystemType**. The key concept here is that a system can be composed of subsystems, which are themselves systems that can be composed of subsystems (etc). A particular aircraft type specification is a tree of nested subsystems, implemented as a recursive “parent” relationship (“pig ear”).

The top level node (which has no parent) is the aircraft type. Parentless nodes are totally unrelated from each other, and the model can hold an arbitrary number of aircraft types. Parentless nodes have no SemVer attribute, because they are abstract.

First generation nodes (with parentless parents) are concrete specifications of their abstract type. They must be semantically labeled with a non-nul SemVer attribute that is unique to the abstract type (but not globally unique, two versions may have the same SemVer attribute value if they are of a different abstract type).

Nodes of “greater than first generation” (whose parents have parents) form a subsystem hirarchy for that version of the abstract type. Where a subsystem has no children, we call it a component. Components and subsystems may optionally have non-nul SemVer attributes (why? because it’s harmless, and potentially useful).


<<abstract type>> Acme Surveywing
<<version>> Acme Surveywing v0.1.4-beta
<<subsystem>> wing
<<component>> left elevon servo
<<component>> right elevon servo
<<component>> Windrider Scout (COTS chevron combat wing)
<<subsystem>> robotic pod
<<component>> fuselage
<<subsystem>> avionics
<<component>> RC reciever
<<component>> autopilot
<<component>> GPS
<<component>> modem
<<subsystem>> payload
<<component>> camera
<<component>> gimbal
<<component>> companion computer

Example Aircraft Type Specification (simplified)

5.1.2 ComponentType and ComponentTypeAppropriateness

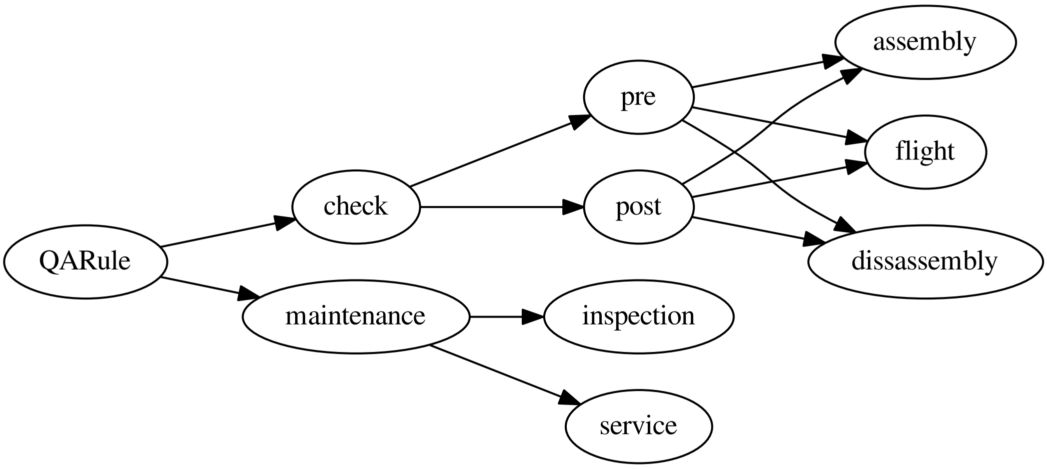
A **ComponentType** is an unambiguous specification of a physical kind of thing, for example a specific make and model of camera.

A **ComponentTypeAppropriateness** is a rule that says a ComponentType explicitly can (or explicitly can not) be employed in the role of a SystemType. For example, a particular make and model of camera is appropriate for the payload subsystem of the robotic pod of an Acme Surveywing v0.1.4-beta.

5.1.3 QARule

Aircraft Types have a set of Quality Assurance Rules (QARules) that drive the behavior of checklist and maintenance systems. These may be bound to the abstract type (e.g. pilot certification), specific component (e.g. specific maintenance requirement) or any subsystem inbetween.

QARules probably form their own type hirarchy, but it requires more analysis. For example, something like this:



QARules would be critical or non-critical. Failing a critical rule prevents takeoff, failing a non-critical rule results in a warning.

Checks would be assembled into checklists that are incorporated into operating procedures.

Maintenance rules would assessed against maintenance logs, resulting in warnings/blocks before flight (preflight checklist integration), post-flight alerts of maintenance falling due as a result of operational activity, as well as fleet management views that indicating upcoming maintenance requirements.

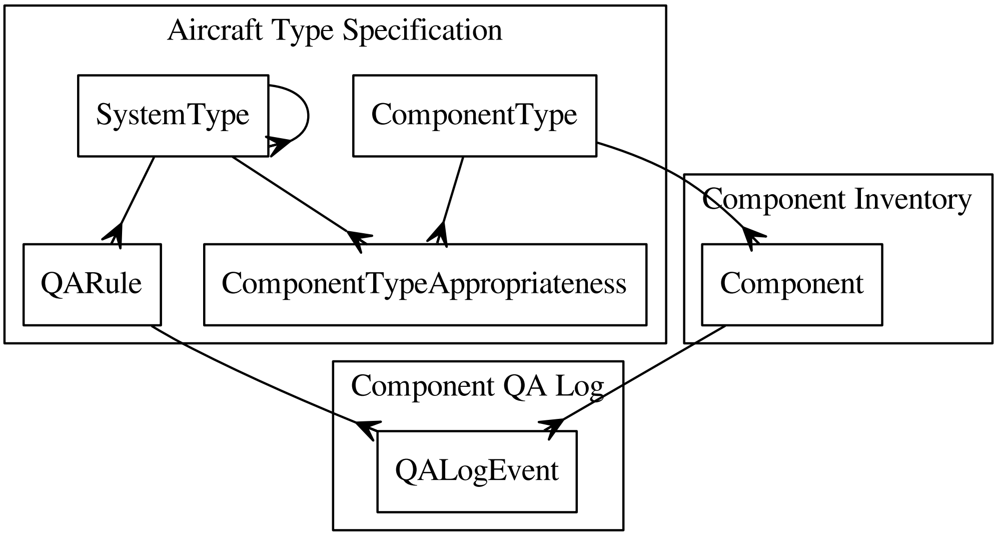
5.2 ComponentInventory

This is like an asset register of uniquely identified physical items. The items are of ComponentType.

Note that it is possible to posess Components (in the ComponentInventory) that are of a ComponentType that is not appropriate for any aircraft type specification. i.e. any type of stuff can be recorded on the asset register, even if it’s not usefull.

5.3 ComponentQALog

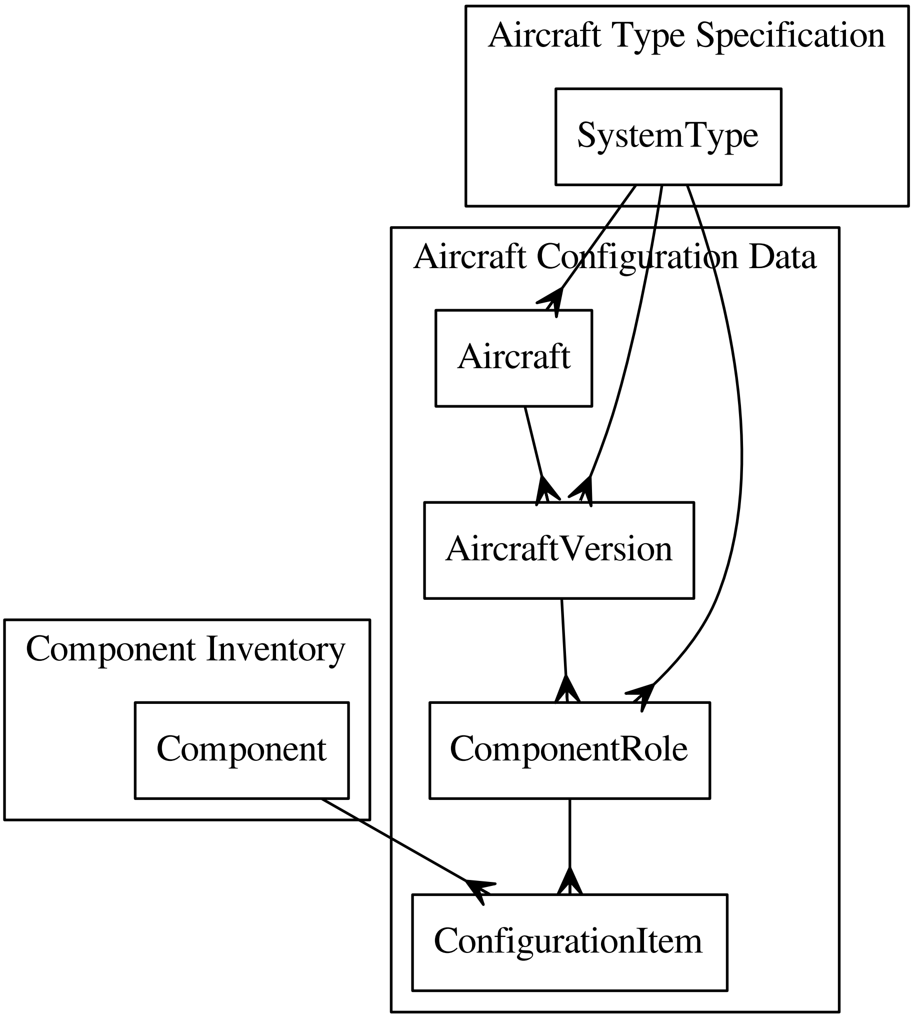
This is a list of things that happened (to Components), which coresponds to QARules being followed.



5.4 Aircraft Configuration Data

This describes a fleet of aircraft. Instances of Aircraft Type Specifications.

An aircraft has a unique identity, but it is an assembly of components that can be swapped out.



5.4.1 Aircraft

A unique identifier. Registration number if you will. Note that it's bound to a SystemType - this must be an abstract type (system with no parents), because an aircraft can not change type!

5.4.2 AircraftVersion

Instance bound to a first generation SystemType. Note that an aircraft can be upgraded (and downgraded) to different versions of it's type.

5.4.3 ComponentRole

Logical entities for a particular version of an aircraft, that corespond to the components (leaves) in the aircraft type specification tree. If the aircraft type specifies a subsystem with a camera component, this specific aircraft's camera is defined as a coresponding ComponentRole.

5.4.4 ConfigurationItem

This represents that a specific component (e.g. camera) is employed in the ComponentRole.

When components are swapped out in an aircraft, this is represented as changes to ConfigurationItems. ConfigurationItems have timestamps ("from" and "to") and there are rules preventing contemporanious assignments of different Components to the same ConfigurationRole (and, the same component to multiple ConfigurationRoles). You can only be in one place at a time.

5.5 Old Stuff

Rude notes from an earlier analysis session...

5.6 HIA (Hazard, Incident, Accident)

5.6.1 HIA Artefacts

Three kinds:

- Report Submission
- Supporting Media
- Supporting URL

5.6.2 HIA Involvement

TODO, elaborate:

- Person
- HIA_Role
- Involvement: HIA_Role –< Involvement >– HIA
- Involved: Person –< Involved >– Involvement

5.7 Safety Team

UAS Operation, the group with the SMS:

- HIA >– Team
- Vehicle >– Team
- TeamRole: Safety Officer, Chief Pilot, etc.
- Team –< TeamInvolvement >– TeamRole
- TeamInvolvement –< TeamInvolved >– Person

Note on Teams:

- system may be configured as multi-tenanted solution (software as a service, multiple teams)
- system may be configured for a single team, "self-hosted" configuration

5.8 Confidentiality

Private: only visible to the Team's Chief Pilot and Safety Officer(s). Note local laws may require the Chief Pilot to report all data on certain accidents to authorities.

Public: may be published, at Chief Pilot or any Safety Officer's discression.

CHAPTER 6

Design Solution

This is also half-baked blather, requiring significant rework before it's usefull. Please focus on requirements for the timebeing.

6.1 Minimum Viable Product

The minimum sufficient technichal solution that would allow the Safety Team to meet the current requirements model is:

- a private email list
- a safety team web page
- one or more document templates (forms) for making HIA submissions

UAS operaters (and anyone else) would be able to submit hazard, incident and accident reports to the private email list. These might be held in quarentine by the mail software (because they are not members of the private list), until such time as a list moderator from the safety team released it to the list.

Safety Team members could communicate with the submitter over email directly, and include the private email list in those messages as required. Supplimentary data could be attached to emails (or referenced with links).

Deficiencies discovered in the mailing list approach could be rectified with engineering effort.

6.2 Possible Future Directions

IMPROVE ACCESSABILITY: Online forms that can be used from web browsers, smart phones, tablets, etc. may have advantages over document-based approaches.

ANALYTIC FEATURES: Storing structured data from submissions in a relational datbase may prove helpful for analysis and reporting.

ADDITIONAL TOOLS: The future safety management system will require more than HIA reporting and analysis, this is just the start.

COMMUNICATION MANAGEMENT: Once we have started systematically improving safety, promoting that capability will become important. Doing this effectively may require development or integration of additional features.

COLABORATE ON TOOLS: If an integrated suite of safety management tools is created, and if those tools are published as open source, then they may be improved as a consequence of broader adoption. This may apply to policies and procedures as well as software.

COLABORATE ON DATA: Opening our non-confidential hazards, incident and accident reports to public scrutiny may improve the quality of the analysis, and subsequently yield better safety outcomes.

HARMONISE ACROSS GROUPS: “multi-tenanted” tools (that host multiple UAS operations) might facilitate information sharing, broader analysis and accelerate the safety improvements for participating groups and the wider community. For example common risk/hazard taxonomies may emerge, relative merits of various mitigations may be debated as a community, etc.

CHAPTER 7

Indices and tables

- *genindex*
- *modindex*
- *search*